

# SB-7SEG8(SensorBlock-7segment LED x8)

## 7セグメント LED I2C ドライバソフト

### ユーザーマニュアル

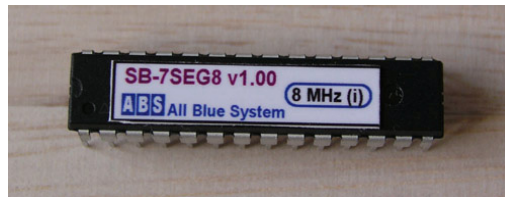
---

ABS-9000 SB-7SEG8

SB-7SEG8 User Manual Rev A.1.03

2012/10/6

Firmware version: “1.00”



オールブルーシステム (All Blue System)

ウェブページ: [www.allbluesystem.com](http://www.allbluesystem.com)

コンタクト: [contact@allbluesystem.com](mailto:contact@allbluesystem.com)

<b>1</b>	<b>このマニュアルについて</b> .....	<b>3</b>
1.1	著作権および登録商標 .....	3
1.2	連絡先 .....	3
<b>2</b>	<b>使用条件およびライセンス</b> .....	<b>3</b>
<b>3</b>	<b>イントロダクション</b> .....	<b>4</b>
<b>4</b>	<b>SB-7SEG8 動作仕様</b> .....	<b>5</b>
4.1	マイクロコントローラ(Atmega168P).....	5
4.2	SB-7SEG8 ファームウェア書き込み時の Fuse Bits 設定値 .....	6
4.3	タイミングと電気特性 .....	6
<b>5</b>	<b>SB-7SEG8 I2Cスレーブコマンドリファレンス</b> .....	<b>6</b>
5.1	自身のスレーブアドレスの設定・取得(0x01) .....	7
5.2	内部タスクインターバル値設定・取得(0x05) .....	7
5.3	LEDフラッシュ表示時間設定・取得(0x06) .....	8
5.4	セグメントカラム数設定・取得(0x07).....	8
5.5	セグメントデータ反転フラグ設定・取得(0x08).....	9
5.6	カラム選択データ反転フラグ設定・取得(0x09).....	9
5.7	LEDの減光パラメータ設定・取得(0x0A) .....	10
5.8	ファームウェアバージョン番号取得(0x0F) .....	10
5.9	セグメントデータ出力(0x10).....	10
5.10	文字データ表示(0x11) .....	11
5.11	セグメントデータクリア(0x12).....	13
<b>6</b>	<b>テスト用回路図</b> .....	<b>14</b>
<b>7</b>	<b>本製品に使用したソフトウェアライセンス表記</b> .....	<b>15</b>
<b>8</b>	<b>サポートについて</b> .....	<b>17</b>
<b>9</b>	<b>更新履歴</b> .....	<b>17</b>

## 1 このマニュアルについて

---

### 1.1 著作権および登録商標

---

Copyright© 2011 オールブルーシステム

このマニュアルの権利はすべてオールブルーシステムにあります。無断でこのマニュアルの一部を複製、もしくは再利用することを禁じます。

Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.

### 1.2 連絡先

---

オールブルーシステム (All Blue System)

ウェブページ <http://www.allbluesystem.com>

メール [contact@allbluesystem.com](mailto:contact@allbluesystem.com)

## 2 使用条件およびライセンス

---

本ソフトウェア(ファームウェア)はオールブルーシステムの ABS-9000 DeviceServer のライセンスを購入されて、その DeviceServer と組み合わせて使用する場合には、複数のプロセッサにインストールして使用することができます。この場合には本ソフトウェアを他の製品に組み込んだり、サポート業務を行うこともできます。

前述の ABS-9000 DeviceServer のライセンスを購入された場合の他に、オールブルーシステムから本ソフトウェアのライセンスを購入された場合に、本ソフトウェア(ファームウェア)の使用ライセンスをお客様に提供いたします。ハードウェアと組み合わせたアプリケーション全体についてのライセンスや、ハードウェアと組み合わせたアプリケーションのサポートは、オールブルーシステムは提供致しません。

本ソフトウェアをオールブルーシステムの ABS-9000 DeviceServer と組み合わせずに単体で使用する場合や、本ソフトウェアライセンスを別途購入していない場合には、個人目的でのみこのソフトウェアを使用することができます。この場合は、業務用途や商用目的で本ソフトウェアを使用したり、他の製品に組み合わせて使用したり、サポート業務をおこなうことはできません。

本ソフトウェアはハイリスクな目的に使用することはできません。ハイリスクな目的とは、原子力、航空、直接的または間接的に人体に死傷を及ぼす可能性のある装置等に使用することを指します。オールブルーシステムは、本ソフトウェアにエラー、バグ等の不具合がないこと、若しくは中断なく稼動すること又は本ソフトウェアの使用がお客様及び第三者に損害を与えないことを保証しません。この事に同意していただけない場合は使用することはできません。

### 3 イントロダクション

センサーブロック SB-7SEG8 ファームウェア(以降“SB-7SEG8”と略す)の機能について説明します。

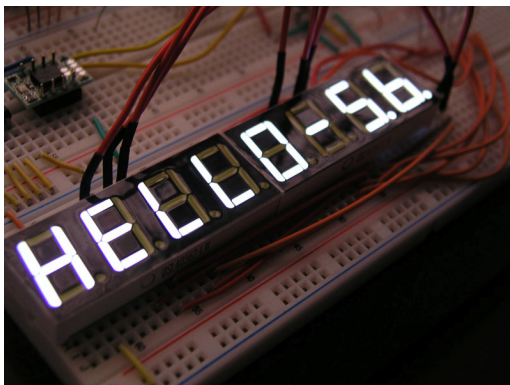
SB-7SEG8 は、Atmel Atmega168P<sup>1</sup> マイクロプロセッサ用のファームウェアです。複数の7セグメントLEDをダイナミックドライブ方式で点灯して、任意のセグメントデータや数値データを表示することができます。SB-7SEG8 は I2C スレーブデバイスとして動作して、I2C マスターデバイスから指定されたデータを LED に出力します。

SB-7SEG8 を使用することで、マスターデバイス側は LED ダイナミック表示のためのタスクから開放されますので、プログラムサイズやCPU 負荷を軽減させることができます。また、SB-7SEG8 ではダイナミック点灯処理を独立して実行できますので、高速(数KHz)に表示切替することでチラツキを低減させています。

SB-7SEG8 は全ての操作を I2C バスを経由したコマンドデータを受信することで実行します。

SB-7SEG8 は使用する 7セグメントLED のカラム数設定や、セグメントやカラム指定ポート値の反転設定などを、I2C 経由のコマンドで設定することができます。SB-7SEG8 の主な機能は、

- I2C マスターデバイスから指定したセグメントパターンを LED に出力することができます。
- I2C マスターデバイスから指定した文字コードデータを LED に出力することができます。
- 最大8桁までの7セグメントLEDをドライブ可能、桁数は1から8まで自由に設定することができます。
- LED 表示データの内容が更新(変更)されたときに、自動的にフラッシュ(点灯と消灯の繰り返し)を数秒間自動的に実行して強調表示することができます。
- 様々な種類の7セグメントLEDやドライブ回路に対応できるように、セグメントデータ出力とカラム選択出力を、各々反転して出力する様に設定することができます。
- LED の明るさを減光するためのパラメータを設定することができます。
- SB-7SEG8 デバイスのI2Cスレーブアドレスを任意の値に変更できます。複数の SB-7SEG8 デバイスを同一I2Cバスに接続して、必要なだけ LED 表示器を増設してマスター側からコントロールすることができます。
- 各種設定値はマイクロプロセッサ内の EEPROM に保存されます。リセット時に自動的にロードされて最後に保存した設定内容で動作します。



<sup>1</sup> Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.

## 4 SB-7SEG8 動作仕様

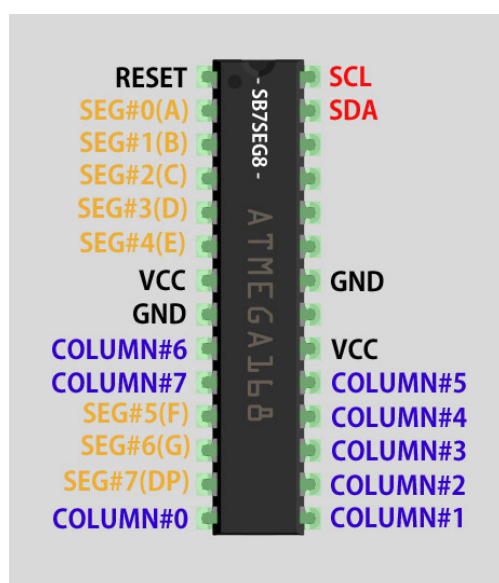
SB-7SEG8 デバイスのファームウェアは下記の動作環境で機能するように設計されています。

- CPU Atmel社製 Atmega168P マイクロコントローラ
- CPU クロック 内部クロック 8MHz
- 電源 5Vまたは3.3V

### 4.1 マイクロコントローラ(Atmega168P)

		設定値
CPU クロック		8MHz 内部クロック
電源電圧		5V または 3.3V
Flash memory		ファームウェア格納用
EEPROM		コンフィギュレーション保存用
PORTB	bit 0-7	セグメントカラム選択 出力ポートとして使用
PORTC	bit 0	リセット直後のSB-7SEG8 起動時に LOW しておくことで、プロセッサ内部のEEPROM コンフィギュレーションデータを初期化することができます。
	bit 1-3	未使用
	bit 4	I2C バスの SDA ピンとして使用 (内部プルアップは無効に設定されています)
	bit 5	I2C バスの SCL ピンとして使用 (内部プルアップは無効に設定されています)
PORTD	bit 0-7	セグメントデータ 出力ポートとして使用

ピン配置図



## 4.2 SB-7SEG8 ファームウェア書き込み時の Fuse Bits 設定値

Fuse Bits	設定値
Low Byte	0b11100010
High Byte (*1)	0b11011100 電源電圧 5V で使用する場合 0b11011101 電源電圧 3.3V で使用する場合
Extended Byte	0b001

(\*1) EEPROM 動作に必要な電源電圧を保障するために、必ず Extended Byte の “BODLEVEL2:0” を上記の値にセットして下さい。

## 4.3 タイミングと電気特性

SB-7SEG8 の電気的な特性は、ファームウェアを搭載するマイクロプロセッサ Atmega168p の仕様に依存しています。詳しくは、Atmel Atmega168P<sup>2</sup> マイクロプロセッサのドキュメントを参照してください。

## 5 SB-7SEG8 I2C スレーブコマンドリファレンス

SB-7SEG8 の I2C スレーブデバイス機能は、I2C マスターデバイスからのコマンドデータを受信してコマンド処理を行った後、コマンドの種類によってはリターン値をマスターデバイスに送信します。

SB-7SEG8 の I2C スレーブデバイスは必ず最初にマスターデバイスから1 byte のコマンドデータと必要に応じてコマンドパラメータを受信します。コマンドデータとパラメータを合わせて最大 32 バイト分のデータをSB-7SEG8 の I2Cスレーブデバイスに送信することができます。

```
<command>, <param#0>, <param#1> ... , <param#30>
```

送信されたコマンドデータごとに予め決められた処理を SB-7SEG8 内で実行して、必要に応じてI2C マスターデバイスに最大 32 バイト分のリターンデータを返します。

```
<return#0>, <return#1>, <return#2> ... <return#31>
```

コマンドデータ送信後にリターンデータをマスターデバイスが受信する場合には、I2C バスを “Master Transmitter” から “Master Receiver” に切り替えて操作して下さい。このとき、先に送信済みのコマンドデータの実行結果をリターン値として取得することができます。

<sup>2</sup> Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.

I2C バス上にある1つのSB-7SEG8 スレーブデバイスに対して、複数のマスターデバイスがアクセスする場合には、それぞれのマスターデバイスは、“Repeated start mode” を使用して上記のモード切り替え操作を行って、コマンド送信からリターン値取得まで、同一のI2C マスターデバイスが I2C バスを占有するようにして下さい。

SB-7SEG8 のコマンド処理は、同一のコマンドバイト値でもコマンドデータ（とコマンドパラメータ）送信後に受信操作を行う場合と、受信操作を行わない場合では動作が異なります。この区別は、最初に I2C マスターから送信されるコマンドデータとコマンドパラメータの内容で判断しています。

次項から、各スレーブコマンドの説明を行います。

## 5.1 自身のスレーブアドレスの設定・取得(0x01)

SB-7SEG8 デバイス自身の現在のスレーブアドレスを取得します。

初期設定値は、0x32 です。

(write) 0x01	(read) <slave_addr> 1 byte
--------------	----------------------------

SB-7SEG8 デバイス自身のスレーブアドレスを設定します。

(write) 0x01	(write) <slave_addr> 1 byte
--------------	-----------------------------

スレーブアドレス設定直後に、新しいスレーブアドレスで SB-7SEG8 デバイスが動作します。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

---

### I2C スレーブアドレスの初期化

---

I2C スレーブアドレスは上記のコマンドで任意の値に設定できますが、間違えて意図しないアドレスに設定された場合に、設定値を初期化することができます。リセット直後のファームウェア起動時に PORTC Bit#0 をLOW にしておくことで、プロセッサ内部のEEPROM コンフィギュレーションデータが全て初期化されます。これによって、スレーブアドレスが初期設定値に戻ります。

---

## 5.2 内部タスクインターバル値設定・取得(0x05)

デバイス内で実行されているタスク(セグメントデータの変化検出)等の実行間隔を取得します。

設定値の単位は ms で通常はデフォルト値 10ms から変更する必要はありません。

初期設定値は、0x0A (10ms) です。

(write) 0x05	(read) <interval> 1 byte
--------------	--------------------------

インターバル値を設定します。(単位は ms)

(write) 0x05	(write) <interval> 1 byte
--------------	---------------------------

インターバル値は変更直後に有効になります。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

### 5.3 LED フラッシュ表示時間設定・取得(0x06)

LED の表示内容が変化したときに、LED をフラッシュ(点灯と消灯を繰り返し)して強調表示させることができます。この設定値を 0 以外の値に設定すると設定した値(秒数)だけフラッシュ表示します。0 を設定するとフラッシュ表示を行いません。

表示の内容が変化した時だけフラッシュしますので、同じセグメントパターンになるデータを連続して更新した場合にはフラッシュ表示されません。

フラッシュ表示時間を取得します。

初期設定値は、0x0 (フラッシュ表示なし) です。

(write) 0x06	(read) <flash_time> 1 byte
--------------	----------------------------

フラッシュ表示する長さを設定します。(単位は sec)

(write) 0x06	(write) <flash_time> 1 byte
--------------	-----------------------------

<flash\_time>は変更直後に有効になります。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

### 5.4 セグメントカラム数設定・取得(0x07)

ダイナミック表示を行う 7セグメントLED のカラム数の設定または取得を行います。

ここで指定したカラム数分のセグメントカラム選択出力信号が PORTB から出力されます。例えば、カラム数が 4 の場合には、PORTB の bit#0 から bit3 にセグメントカラム選択信号が出力されます。

カラム数を取得します。

初期設定値は、0x08 です。



(write) 0x07	(read) <column> 1 byte
--------------	------------------------

カラム数を設定します。

(write) 0x07	(write) <column> 1 byte
--------------	-------------------------

<column>は変更直後に有効になります。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.5 セグメントデータ反転フラグ設定・取得(0x08)

ダイナミック表示を行う時に、7セグメントLEDに出力するセグメントデータ値を反転するかどうかのフラグを、設定また取得します。アノードコモンタイプの 7セグメントLED を使用する場合に 0x01 を設定します。

フラグ値を取得します。

初期設定値は、0x0 です。

(write) 0x08	(read) <seg_data_inv> 1 byte
--------------	------------------------------

フラグ値を設定します。フラグ値は 0x0 で無効、0x01 で有効に設定されます。

(write) 0x08	(write) <seg_data_inv> 1 byte
--------------	-------------------------------

<seg\_data\_inv>は変更直後に有効になります。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.6 カラム選択データ反転フラグ設定・取得(0x09)

ダイナミック表示を行う時に、表示対象の7セグメントLEDカラムを選択するためのポート値を反転するかどうかのフラグを、設定また取得します。カソードコモンタイプの 7セグメントLED を使用する場合に 0x01 を設定します。

フラグ値を取得します。

初期設定値は、0x01 です。

(write) 0x09	(read) <seg_select_inv> 1 byte
--------------	--------------------------------

フラグ値を設定します。フラグ値は 0x0 で無効、0x01 で有効に設定されます。

(write) 0x09	(write) <seg_select_inv> 1 byte
--------------	---------------------------------

<seg\_select\_inv>は変更直後に有効になります。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.7 LEDの減光パラメータ設定・取得(0x0A)

7セグメントLEDの明るさを調整するためのパラメータを、設定または取得します。

パラメータ値が 0 の場合にLED が一番明るくなり、パラメータ値が大きくなるにしたがって暗くなります。

減光パラメータ値を取得します。

初期設定値は、0x0 です。

(write) 0x0A	(read) <dimmer_value> 1 byte
--------------	------------------------------

減光パラメータ値を設定します。減光パラメータ値は 0x0 から 0xff までの値が有効です。

(write) 0x0A	(write) <dimmer_value> 1 byte
--------------	-------------------------------

<dimmer\_value>は変更直後に有効になります。

ここで設定した内容は SB-7SEG8 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.8 ファームウェアバージョン番号取得(0x0F)

ファームウェアバージョン番号を取得します。

(write) 0x0F	(read) <version> 1 byte
--------------	-------------------------

<version> は 0x64 から 0xFF までの値で、整数を実数に変換して100 で割った値がバージョン番号となります。

例: <version> の値が 0x65 の場合には ver 1.01

## 5.9 セグメントデータ出力(0x10)

7セグメント LED の各カラムのセグメントに出力されるデータを指定します。

カラムは左から順番に指定して、連続して最大 8カラム分のセグメントデータを指定できます。指定しなかったカラムのセグメントデータは変化しません。

全てのカラムのセグメントデータ初期値は、0x00 です。

(write) 0x10	(write) <seg1> 1 byte	(write) <seg2> 1 byte	(write) <seg3> 1 byte
--------------	-----------------------	-----------------------	-----------------------

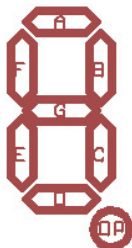
(続く)

(write) <seg4> 1 byte	(write) <seg5> 1 byte	(write) <seg6> 1 byte	(write) <seg7> 1 byte
-----------------------	-----------------------	-----------------------	-----------------------

(続く)

(write) <seg8> 1 byte
-----------------------

セグメントデータのパターンは、下記のセグメント(A..G とDP) が 1バイト<seg(n)> 中の各ビットに対応しています。



<seg(n)> データとセグメントの対応。対応するビット値が 1 の場合に点灯を意味します。

MSB(bit7)	(bit6)	(bit5)	(bit4)	(bit3)	(bit2)	(bit1)	LSB(bit0)
DP	G	F	E	D	C	B	A

例えば、セグメントカラムの1桁目から3桁目までアルファベットの ‘A’ を表示する場合に、I2C マスタから送信する I2C データはコマンドバイトを含めて、“0x10,0x77,0x77,0x77” になります。

## 5.10 文字データ表示(0x11)

7セグメント LED の各カラムに表示するデータを文字コードまたはバイナリ値で指定します。

カラムは左から順番に指定して、連続して最大 8カラム分の文字データを指定できます。指定しなかったカラムの表示内容は変化しません。

表示する文字コードは、ASCII で ‘0’ から ‘9’ まで(0x30 .. 0x39) と ‘A’ から ‘F’ まで(0x41 .. 0x46) のデータをセグメントパターンに変換して表示します。それ以外の文字コードを指定した場合でも、できるだけ対応するセグメントパターンに変換して表示します。変換するセグメントパターンが存在しない文字はブランクになります。

文字コードに 0x00 から 0x0F を指定した場合には、そのバイナリ値を表現するセグメントパターンに変換して表示します。例えば、0x00 を指定した場合には、0x30 を指定した場合と同様に ‘0’ を 7セグメントLED に表示します。

‘.’ ペリオドは、7セグメントLED 中の “DP” セグメントを点灯させます。文字データ中のペリオドは 1つ前のカラムの “DP” セグメントが点灯します。

(write) 0x11	(write) <character_datas> max 31 byte
--------------	---------------------------------------

例えば、<character\_datas> に “0x33, 0x2E, 0x31, 0x34, 0x31, 0x0a, 0x0b, 0x0c, 0xff” を指定すると、7セグメントLEDには “3.141AbC “ が表示されます。このとき小数点は第一セグメントに表示され、第8セグメントは変換するパターンが存在しないため空白表示になります。

文字データ ASCII またはバイトデータ (0x??)	点灯セグメント
0x00, '0', '0'	(A) (B) (C) (D) (E) (F)
0x01, '1', '1'	(B) (C)
0x02, '2'	(A) (B) (D) (E) (G)
0x03, '3'	(A) (B) (C) (D) (G)
0x04, '4'	(B) (C) (F) (G)
0x05, '5', 'S', 's'	(A) (C) (D) (F) (G)
0x06, '6'	(A) (C) (D) (E) (F) (G)
0x07, '7'	(A) (B) (C) (F)
0x08, '8'	(A) (B) (C) (D) (E) (F) (G)
0x09, '9', 'g'	(A) (B) (C) (D) (F) (G)
0x0a, 'A', 'a'	(A) (B) (C) (E) (F) (G)
0x0b, 'B', 'b'	(C) (D) (E) (F) (G)
0x0c, 'C'	(A) (D) (E) (F)
'c'	(D) (E) (G)
0x0d, 'D', 'd'	(B) (C) (D) (E) (G)
0x0e, 'E'	(A) (D) (E) (F) (G)
'e'	(A) (B) (D) (E) (F) (G)
0x0f, 'F', 'f'	(A) (E) (F) (G)
'G'	(A) (C) (D) (E) (F)
'H'	(B) (C) (E) (F) (G)
'h'	(C) (E) (F) (G)
'I', 'i'	(B) (C)
'J', 'j'	(B) (C) (D) (E)
'L'	(D) (E) (F)
'N', 'n'	(C) (E) (G)
'o'	(C) (D) (E) (G)
'P', 'p'	(A) (B) (E) (F) (G)
'Q', 'q'	(A) (B) (C) (F) (G)
'R', 'r'	(E) (G)
'T', 't'	(D) (E) (F) (G)
'U'	(B) (C) (D) (E) (F)
'u'	(C) (D) (E)

'Y', 'y'	(B) (C) (D) (F) (G)
'_'	(G)
''''	(B) (F)
0x27	(B)
'_'	(D)
'#' °C のつもり	(A) (F) (G)

## 5.11 セグメントデータクリア(0x12)

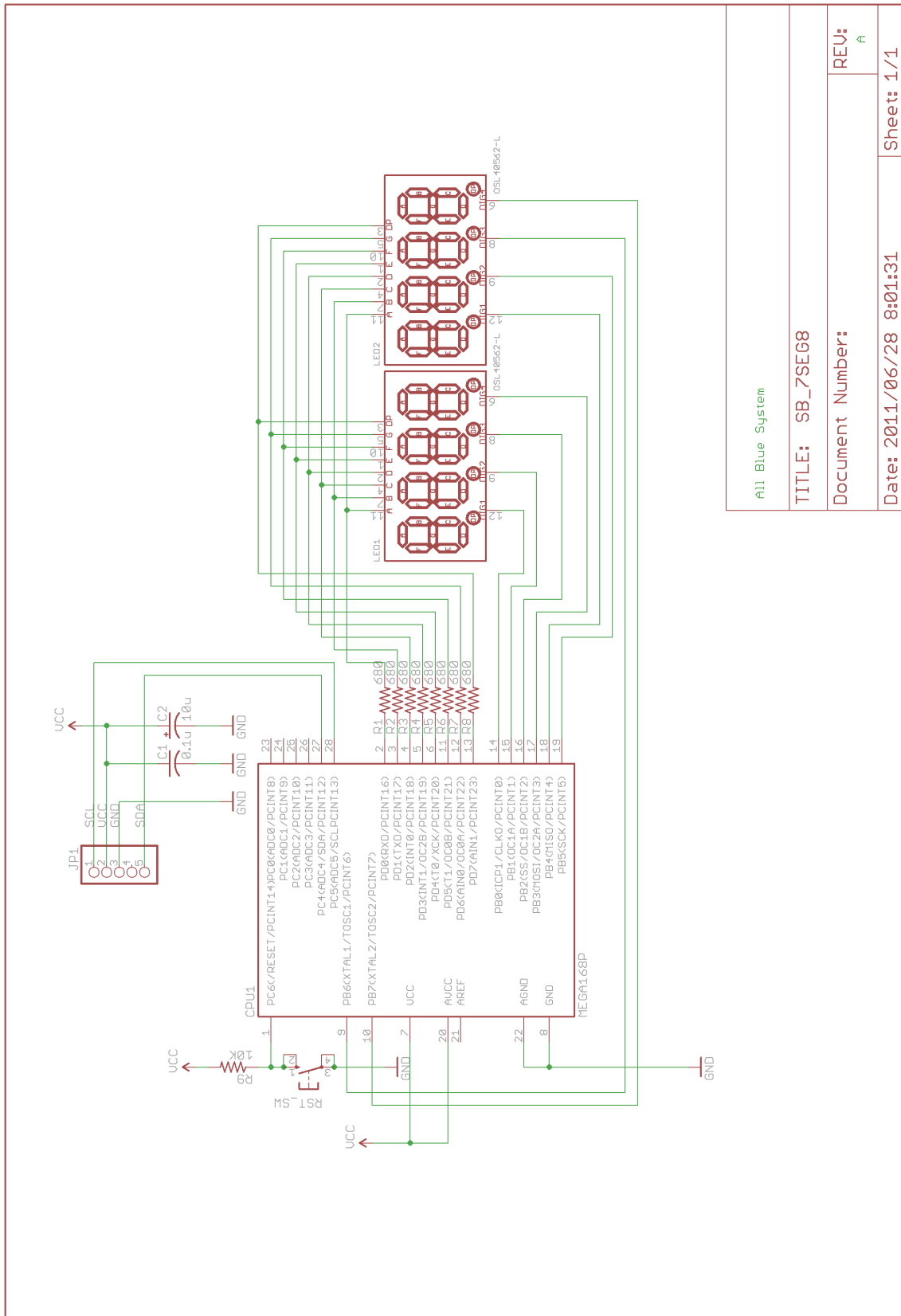
---

全てのカラムの 7セグメント LED をクリアします。

(write) 0x12

## 6 テスト用回路図

テストは、カソード共通タイプの高輝度4連7セグメントLED を2つ使用しています。SB-7SEG8 のコンフィギュレーションを設定することで、アノード共通タイプやセグメントカラム数を変更した回路にも柔軟に対応できます。



All Blue System

TITLE: SB\_7SEG8

Document Number:

REV: A

Date: 2011/06/28 8:01:31

Sheet: 1/1

## 7 本製品に使用したソフトウェアライセンス表記

---

### avr-libc License

avr-libc can be freely used and redistributed, provided the following license conditions are met.

Portions of avr-libc are Copyright (c) 1999-2008

Werner Boellmann,

Dean Camera,

Pieter Conradie,

Brian Dean,

Keith Gudger,

Wouter van Gulik,

Bjoern Haase,

Steinar Haugen,

Peter Jansen,

Reinhard Jessich,

Magnus Johansson,

Harald Kipp,

Carlos Lamas,

Cliff Lawson,

Artur Lipowski,

Marek Michalkiewicz,

Todd C. Miller,

Rich Neswold,

Colin O'Flynn,

Bob Paddock,

Andrey Pashchenko,

Reiner Patommel,

Florin-Viorel Petrov,

Alexander Popov,

Michael Rickman,

Theodore A. Roth,

Juergen Schilling,

Philip Soeberg,

Anatoly Sokolov,

Nils Kristian Strom,

Michael Stumpf,

Stefan Swanepoel,  
Helmut Wallner,  
Eric B. Weddington,  
Joerg Wunsch,  
Dmitry Xmelkov,  
Atmel Corporation,  
egnite Software GmbH,

The Regents of the University of California.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



## 8 サポートについて

---

SB-7SEG8 製品のファームウェアをオールブルーシステムの DeviceServer と組み合わせて使用する場合には、ABS-9000 DeviceServer のライセンスのサポート期間内において、メールにてサポートを行います。

オールブルーシステムの SB-7SEG8 はファームウェアとそのソフト的な不具合に対してのみサポート致します。お客様のハードウェアへの組み込みに関する内容や、マイクロコントローラ自身に関しましてはサポートできません。本ソフトウェアをオールブルーシステムの DeviceServer と組み合わせずに、使用する場合はサポートは行いません。

ABS-9000 DeviceServer のスクリプトライブラリ関数(Luaのライブラリ)とEXCEL VBA から利用するための API ライブラリ関数(XASDLCMD.DLL)を利用する場合の、プログラミング方法やサンプルプログラムのソースコードに関する解説など、お客様のソフトウェア開発自身に関するサポートにつきましては、別途有償対応となります、詳しくはお問い合わせください。

動作環境を満たしている場合でも、お客様の環境やハードウェアによっては正常に動作しない場合があります。これらを含めて、こちらで再現できない問題については十分なサポートができない場合があります。ABS-9000 DeviceServer のライセンスをご購入になる前に、目的の機能がお客様の環境で動作することを、事前にデモライセンスをご利用になって、充分確認されることをお勧めします。

## 9 更新履歴

---

REV A. 1. 03 2012/10/6

ピン配置図 VCC (AVcc) を追加した

REV A. 1. 02 2011/12/5

I2C スレーブアドレス初期化の説明を追加

REV A. 1. 01 2011/11/2

ピン配置図を追加した

文字データからセグメントパターンへの変換表を追加した

REV A. 1. 00 2011/7/5

初版作成